Modeling and Compositional Analysis of Genetic Regulatory Networks

Gregor Gössler

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Modeling and Compositional Analysis of Genetic Regulatory Networks

Gregor Gössler

N° 5801

Janvier 2006

Thème COM

ISSN 0249-6399 ISRN INRIA/RR--5801--FR+ENG

INRIA
RHÔNE-ALPES

Modeling and Compositional Analysis of Genetic Regulatory Networks

Gregor Gössler

Thème COM — Systèmes communicants
Projet Pop Art

Rapport de recherche n° 5801 — Janvier 2006 — 18 pages

Abstract: Proteins fulfil a huge number of functions in any living organism. The dynamics of the protein concentrations in a cell is defined by a regulatory network which usually encompasses a multitude of highly complex feedback loops. Being able to model and analyze its structure and behavior is crucial for understanding the functions of the proteins and their interactions.

For a modeling framework to be useful in practice, two factors are crucial: the model must faithfully represent the actual behavior of the network, and it must be supported by analysis algorithms that are efficient enough to cope with complex models, and scale up well. Therefore, this paper makes two different contributions.

First, we present a framework for modeling genetic regulatory networks in a modular yet faithful manner based on the mathematically well-founded formalism of differential inclusions. In our approach, the different components of the system (proteins or sets proteins) and the way they constrain each other, are modeled separately and modularly.

Second, we propose efficient algorithms to analyze the behavior of the model. The algorithms are compositional in the sense that they verify local properties on the individual components. Sufficient conditions on these local properties and on the constraints between the components then allow to efficiently infer properties of the whole system, such as reachability and existence of equilibrium states.

Two case studies show the potential of this approach.

Key-words: gene network, differential inclusions, modularity, compositional analysis, equilibria, reachability

Unité de recherche INRIA Rhône-Alpes
655, avenue de l'Europe, 38334 Montbonnot Saint Ismier (France)
Téléphone : +33 4 76 61 52 00 — Télécopie +33 4 76 61 52 52

Modélisation et analyse compositionnelle de réseaux géniques

Résumé : Les protéines assurent un grand nombre de fonctions dans tout organisme vivant. La dynamique des concentrations de protéines dans une cellule est définie par un réseau régulatoire qui contient souvent de nombreuses boucles de rétroaction complexes. Pour comprendre les fonctions des protéines et leurs interactions, il est nécessaire de modéliser et analyser sa structure et son comportement.

Pour qu'un cadre de modélisation soit utile en pratique, deux facteurs sont primordiaux : le modèle doit représenter fidèlement le comportement réel du réseau, et il doit être supporté par des algorithmes d'analyse qui soient assez efficaces pour traiter des modèles complexes et qui passent à l'échelle. Dans cet objectif, l'article propose deux contributions.

D'abord, nous présentons un cadre pour la modélisation modulaire de réseaux géniques qui est basé sur le formalisme bien fondé des inclusions différentielles. Dans notre approche, les différents composants du système (protéines ou ensembles de protéines) et la manière dont ils se contraignent mutuellement, sont modélisés séparément et de façon modulaire.

Ensuite, nous proposons des algorithmes efficaces pour analyser le comportement du modèle. Ces algorithmes sont compositionnels dans le sens où ils vérifient des propriétés locales sur les composants individuels. Des conditions suffisantes sur ces propriétes locales et sur les contraintes entre les composants permettent alors d'inférer efficacement des propriétés du système global, telles que l'atteignabilité et l'existence d'équilibres.

Deux études de cas montrent le potentiel de cette approche.

Mots-clés : réseau génique, inclusions différentielles, modularité, analyse compositionnelle, états d'equilibre, atteignabilité

1 Introduction

Genetic regulatory networks. Proteins fulfil a huge number of functions in any living organism. Any protein is encoded by a gene. In order to produce the protein, the corresponding gene has to be *transcribed* into messenger RNA (mRNA), which is then *translated* to obtain the protein [21]. This production mechanism is regulated by the concentration of proteins, which can *activate* or *inhibit* the production, e.g. by binding to the gene and disabling transcription. At the same time, proteins are *degraded*. The behavior of a genetic network can be modeled by a system of differential equations of the form

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}) - \mathbf{g}(\mathbf{x}, \mathbf{u})\mathbf{x} \tag{1}$$

where $\mathbf{x}$ is a vector of protein concentrations representing the current state, $\mathbf{u}$ is a vector of *input* concentrations, and $\mathbf{f}$ and $\mathbf{g}$ model the production rates, and degradation rates, respectively [13]. The dynamics of the protein concentrations is thus defined by a regulatory network which usually encompasses a multitude of highly complex feedback loops. Being able to analyze its structure and behavior is crucial for understanding the functions of the proteins and their interactions.

Modeling with piecewise linear differential equations. The approach of [13] considers an abstraction of the continuous state space into a set of qualitative states, in each of which (1) is approximated with a system of linear differential equations. More precisely, the state space of every variable x_i is partitioned into a set of intervals

$$\mathcal{D}_i^r = \{[0, \theta_i^1)\} \cup \{(\theta_i^j, \theta_i^{j+1}) \mid 1 \leqslant j < p_i\} \cup \{(\theta_i^{p_i}, max_i]\}$$

and a set of threshold values $\mathcal{D}_i^s = \{\theta_i^j \mid 1 \leqslant j \leqslant p_i\}$, where $\theta_i^j < \theta_i^{j+1}$ for $1 \leqslant j \leqslant p_i$. Let $\mathcal{D}_i = \mathcal{D}_i^r \cup \mathcal{D}_i^s$. The global state space is thus partitioned into *domains* $\mathcal{D} = \mathcal{D}_1 \times \mathcal{D}_2 \times ... \times \mathcal{D}_n$. The domains in $\mathcal{D}^r = \mathcal{D}_1^r \times \mathcal{D}_2^r \times ... \times \mathcal{D}_n^r$ are called *regulatory* domains, the domains $\mathcal{D}^s = \mathcal{D} \setminus \mathcal{D}^r$ are called *switching domains*.

In regulatory domains $D \in \mathcal{D}^r$, the concentration varies according to the ratio of production rate and degradation rate:

$$\dot{\mathbf{x}} = \boldsymbol{\mu}^D - \boldsymbol{\nu}^D \mathbf{x} \tag{2}$$

with equilibrium state $\boldsymbol{\phi}$ such that

$$\phi_i(D) = \mu_i^D / \nu_i^D$$

If $\boldsymbol{\phi}(D) \in D$ then the systems stays in D, otherwise it eventually leaves D and enters an adjacent switching domain. In switching domains, the evolution of the system cannot be computed in the same way, since $\boldsymbol{\mu}$ and $\boldsymbol{\nu}$ are not defined there. A solution to this problem is the use of differential inclusions as proposed by Filippov [16]. Depending on the behavior in the adjacent regulatory domains, switching domains are either crossed instantaneously, or the system may stay in the switching domain for some time.

The continuous model of (1) can be conservatively approximated by a discrete transition system on the set of qualitative states $\mathcal{D}$ [13]. By exploring this transition system it is possible to simulate the behavior of the underlying genetic network [11], or verify properties like bistability [3]. However, this method is currently limited by the exponential growth of the transition system with the number of proteins. Since properties are analyzed by enumeratively exploring the state space from an initial state, states that are not reachable from the initial state are not taken into account.

Modularity and compositional analysis. The intrinsic complexity of interactions between genes, proteins, metabolism and other subsystems considered in systems biology, has recently led researchers to study the natural modularity of interactions and to reflect it in their models, so as to be able to provide more efficient algorithms [23].

Compositional analysis means that the behavior of a system consisting of different components (in our case, proteins or sets of proteins) is analyzed by examining the behavior of the components and how they interact, rather than by analyzing the behavior of the overall system. Roughly speaking, the compositional approach only has to deal with the sum of the state spaces of the components — which is usually much smaller than the state space of the whole system —, such that it can be more efficient than non-compositional analysis, and scale better. Its downside is the loss of information due to abstraction, which may lead to less accurate results.

A precondition for compositional algorithms to be applicable, is that the model be structured. Therefore, this paper actually deals with two different issues: modeling genetic regulatory networks in a modular yet faithful manner, and compositionally analyzing the behavior of the model. We propose a modeling framework for genetic regulatory networks. The different components of the system (proteins or sets proteins) and the way

they constrain each other, are modeled separately and modularly. The verification algorithms are compositional in the sense that they verify local properties on the individual components. Sufficient conditions on these local properties and on the constraints between the components then allow to efficiently infer properties of the whole system, such as reachability or the existence of equilibrium states.

Related work. By now there is a large number of approaches to model and analyse genetic networks. A good overview is given in the survey of [10]. The modeling approaches adopt different mathematical frameworks, which vary in expressiveness and the availability and efficiency of verification algorithms. Most of the algorithms "flatten" the model and work on the global state space, without computationally taking advantage of the modularity of the problem.

There is a wide variety of modeling approaches based on differential equations. However, simulation and verification of the continuous model can be expensive, and many properties are not even decidable in this framework. Therefore, several ways have been investigated to discretize the continuous model defined by differential equations while preserving behavioral properties [17], and more specifically soundness [13] and reachability [4]. [2] uses predicate abstraction to automatically find a conservative approximation of reachability in a linear hybrid system.

In order to deal with complex networks, it may be a good choice to change precision against efficiency, and directly model genetic networks in a discrete framework, such as systems of discrete [6, 20] or Boolean [8, 9] functions, or rule-based formalisms like term rewriting systems [14, 15].

Organization of the paper. In Section 2, we introduce the modeling framework. Starting from the qualitative model of [13], we show show how a genetic network can be modeled in a modular way in our framework, and compare both models. Section 3 presents symbolic verification algorithms that exploit the modularity of the model. Section 4 illustrates our results with two case studies, and Section 5 concludes.

2 Component-based Modeling

2.1 Components and Constraints

In the following, we present a simplified version of the component model adopted in [19]. For a set of variables X, let $\mathbf{X}$ denote the set of valuations of X, and let $\mathcal{P}(X) = 2^{\mathbf{X}}$ be the set of predicates on $\mathbf{X}$.

Definition 1 (Transition system) *A* transition system B *is a tuple* (X, A, A^c, G, F) *where*

- X *is a finite set of variables;*
- A *is a finite set of actions, union of two disjoint sets* A^u *and* A^c*, the sets of the* uncontrollable *and* controllable *actions, respectively;*
- $G : A \to \mathcal{P}(X)$ *associates with every action its* guard *specifying when the action can occur;*
- $F : A \to (\mathbf{X} \to \mathbf{X})$ *associates with every action its* transition function.

Uncontrollable actions are used to represent input events that cannot be triggered nor prevented by the modeled system. For convenience, we write G^a and F^a for $G(a)$ and $F(a)$, respectively.

Definition 2 (Semantics of a transition system) *A transition system* $B = (X, A, A^c, G, F)$ *defines a* transition relation $\to$: $\mathbf{X} \times A \times \mathbf{X}$ *such that:* $\forall \mathbf{x}, \mathbf{x}' \in \mathbf{X}\ \forall a \in A\ .\ \mathbf{x} \xrightarrow{a} \mathbf{x}' \iff G^a(\mathbf{x}) \wedge \mathbf{x}' = F^a(\mathbf{x})$.

Notations: as usual, we write $\mathbf{x} \to \mathbf{x}'$ for $\exists a \in A\ .\ \mathbf{x} \xrightarrow{a} \mathbf{x}'$, and $\to^*$ for the transitive and reflexive closure of $\to$. Given $B = (X, A, A^c, G, F)$, an action $a \in A$ is *reachable* from some state $\mathbf{x}$ if $\exists \mathbf{x}', \mathbf{x}''\ .\ \mathbf{x} \to^* \mathbf{x}' \wedge x' \xrightarrow{a} \mathbf{x}''$. We represent by $G(B)$ the disjunction of its guards, that is $G(B) = \bigvee_{a \in A} G^a$.

Definition 3 (Predecessors) *Given a transition system* $B = (X, A, A^c, G, F)$ *and a predicate* $P \in \mathcal{P}(X)$*, the* predecessors *of* P *by action* a *is the predicate* $pre_a(P) = G^a \wedge P[F^a(X)/X]$ *where* $P[F^a(X)/X]$ *is obtained from* P *by uniform substitution of* X *by* $F^a(X)$.

Clearly, $pre_a(P)$ characterizes all the states from which execution of a leads to some state satisfying P.

Definition 4 (Controllable predecessors) *Given $B = (X, A, A^c, G, F)$ and $P \in \mathcal{P}(X)$, the predicate $pre\_c(P) = \bigvee_{a \in A^c} pre_a(P) \wedge \neg \bigvee_{a \in A^u} pre_a(\neg P)$ characterizes the* controllable predecessors *of P in one step.*

Let $pre\_c^0(P) = P$, and $pre\_c^{i+1}(P) = pre\_c\big(pre\_c^i(P)\big)$ be the controllable predecessors of P in $i \geqslant 0$ steps.

Let $PRE(P)$ be the least solution of $Y = P \cup pre\_c(Y)$, that is, the controllable predecessors of P in an arbitrary number of steps.

Intuitively, P can be reached from $pre\_c(P)$ by some controllable action, independent of the occurrence of uncontrollable actions.

Definition 5 (Invariant) *Given $B = (X, A, A^c, G, F)$ and a predicate $P \in \mathcal{P}(X)$, P is an* invariant *of B if $P \implies \bigwedge_{a \in A} \neg pre_a(\neg P)$. An invariant $P \neq false$ is called* deadlock-free *if $P \implies G(B)$.*

We define two operations on components: composition and restriction. The restriction of a component is a component again, and so is the composition of components.

Definition 6 (Composition) *Let $B_i = (X_1, A_i, G_i, F_i)$, $i = 1, 2$, with $X_1 \cap X_2 = \emptyset$ and $A_1 \cap A_2 = \emptyset$. $B_1 \| B_2$ is defined as the transition system $(X_1 \cup X_2, A_1 \cup A_2, G_1 \cup G_2, F_1 \cup F_2)$.*

This is the standard asynchronous product.

2.2 Restriction

Restrictions allow to constrain the behavior of a transition system.

Definition 7 (State constraint) *Given a transition system $B = (X, A, A^c, G, F)$, a* state constraint *is a predicate $P \in \mathcal{P}(X)$.*

Definition 8 (Action constraint) *Given a transition system $B = (X, A, A^c, G, F)$, an* action constraint *is a tuple of predicates $V = (U^a)_{a \in A^c}$.*

Definition 9 (Restriction) *The* restriction *of $B = (X, A, A^c, G, F)$ with a state constraint P and an action constraint $V = (U^a)_{a \in A^c}$ is the transition system $B/(P, V) = (X, A, A^c, G', F)$ where for any $a \in A^c$, $(G^a)' = G^a \wedge U^a \wedge pre_a(P)$ is the (restricted) guard of a in B/V. The guards of uncontrollable actions remain unchanged.*

Notations: We write B/V for $B/(true, V)$. Let $blocked(a) = G^a \wedge \neg(G^a)'$, and let $EA(\mathbf{x}) = \{a \in A \mid (G^a)'(\mathbf{x})\}$ be the set of actions that are enabled at $\mathbf{x}$. Given a predicate P on an n-dimensional domain $\mathcal{D}$ and $i \in \{1, ..., n\}$, let $inc_i(P) = P[x_i - 1/x_i]$, and $dec_i(P) = [x_i + 1/x_i]$ denote the predicate P "shifted" by one along the i-th dimension, towards higher and lower values, respectively. For example, $inc_1(x_1 = 2 \wedge x_2 > 3) = (x_1 = 3 \wedge x_2 > 3)$.

As a special case of action constraint, priorities have been shown to be *composable deadlock-free control invariants* [1]. In other words, they allow to ensure safety properties in a modular way, while preserving deadlock freedom.

Definition 10 (Priority) *A* priority *on a set of actions $A = A^c \cup A^u$ is a relation $\prec \subseteq A^c \times A$.*

Notations: for $A' \subseteq A$, let $A'/\prec = \{a \in A' \mid \neg \exists a' \in A' \,.\, a \prec a'\}$ be the set of maximal actions in A'. Given sets of actions A_1, A_2 we write $A_1 \prec A_2$ for $\forall a_1 \in A_1 \, \forall a_2 \in A_2 \,.\, a_1 \prec a_2$.

A *dynamic priority* is a function associating a priority with every state of B.

Definition 11 (Dynamic priority) *A* dynamic priority *on a transition system $B = (X, A, A^c, G, F)$ is a tuple of predicates $pr = (C_{ij})_{a_i \in A^c, a_j \in A}$. The action constraint defined by pr, $V(B, pr) = (U^a)_{a \in A^c}$ is $U^{a_i} = \bigwedge_{a_j \in A} \neg(C_{ij} \wedge G^{a_j})$.*

The predicates C_{ij} specify priority between actions a_i and a_j. If C_{ij} is true at some state, then in the system restricted by *pr* the action a_i cannot be executed if a_j is enabled. We write $C_{ij} \mapsto a_i \prec a_j$ to express the fact that a_i is dominated by a_j when C_{ij} holds. Given a dynamic priority *pr* and a state $\mathbf{x}$, let $pr(\mathbf{x}) = \{a_i \prec a_j \mid C_{ij}(\mathbf{x})\}$ be the priority defined by *pr* at $\mathbf{x}$. A dynamic priority is *irreflexive* if $C_{ij} \implies \neg C_{ji}$ for all $(a_i, a_j) \in A^c \times A$.

Definition 12 (Transitive closure) *Given a priority $\prec$ we denote by $\prec^+$ the transitive closure of $\prec$. Given a dynamic priority $pr = (C_{ij})_{a_i \in A^c, a_j \in A}$ we denote by pr^+ the least dynamic priority such that $\forall i,j,k \,.\, C_{ij} \wedge C_{jk} \implies C_{ik}$.*

Definition 13 (Composition $\oplus$ of priorities) *The composition of two priorities $\prec^1$ and $\prec^2$ is $\prec^1 \oplus \prec^2 = (\prec^1 \cup \prec^2)^+$. The composition of two dynamic priorities pr^1 and pr^2 with $pr^k = (C^k_{ij})_{a_i \in A^c, a_j \in A}$, $k = 1,2$, is $pr^1 \oplus pr^2 = \big((C^1_{ij} \vee C^2_{ij})_{a_i \in A^c, a_j \in A}\big)^+$.*

2.3 Component-based Modeling of Genetic Networks

We make two fundamental distinctions on actions and the state variables they modify. First, there is a partition into constrained and unconstrained variables. Second, both are partitioned according to their controllability.

Constrained vs. unconstrained variables. Unconstrained (or input) variables are not subject to equilibria. They can be considered as constant as in [12], in which case they can be eliminated before applying the verification algorithms. We assume here that unconstrained variables may evolve like constrained variables.

Controllability. The distinction between controllable and uncontrollable actions allows to consider the system as a *two-player game* between a *controller* and the *environment*. Uncontrollable actions are triggered by the environment and cannot be enforced or prevented by the controller, whereas controllable actions can be triggered or disabled by a controller so as to satisfy some property (for instance, reach some state). If the controller can satisfy the property however the environment behaves, it has a *winning strategy*.

This partition is orthogonal to the partition into constrained and unconstrained variables. Which actions are chosen to be (un)controllable depends on the property to be verified. For instance, in order to answer the question "how to control input concentrations in order to reach some state", only the unconstrained actions are considered as controllable. If the problem is whether some state is reachable at all, then all actions should be considered as controllable. If we are interested in whether some state is reachable in a hostile environment, then exactly the input actions are to be considered as uncontrollable.

2.3.1 Piecewise linear systems

Definition 14 (Domains) *Consider a Cartesian product $\theta = \theta_1 \times ... \times \theta_n$ with $\theta_i = \{\theta_i^1, ..., \theta_i^{p_i}\}$ an ordered set of* thresholds*, such that $0 < \theta_i^1 < ... < \theta_i^{p_i}$. Let*

$$\mathcal{D}^r_i(\theta) = \{[0, \theta_i^1)\} \cup \{(\theta_i^j, \theta_i^{j+1}) \mid 1 \leqslant j < p_i\} \cup \{(\theta_i^{p_i}, \infty)\}$$

and $\mathcal{D}^s_i(\theta) = \{\theta_i^j \mid 1 \leqslant j \leqslant p_i\}$. We omit the argument θ when it is clear from the context. Let $\mathcal{D}_i = \mathcal{D}^r_i \cup \mathcal{D}^s_i$. The state space $\mathbb{R}^n_{\geqslant 0}$ is thus partitioned into domains *$\mathcal{D} = \mathcal{D}_1 \times \mathcal{D}_2 \times ... \times \mathcal{D}_n$. The domains in $\mathcal{D}^r = \mathcal{D}^r_1 \times \mathcal{D}^r_2 \times ... \times \mathcal{D}^r_n$ are called* regulatory *domains, the domains $\mathcal{D}^s = \mathcal{D} \setminus \mathcal{D}^r$ are called* switching domains*.*

Definition 15 (Piecewise linear system) *Let $M = (X, U, Y, \theta, \boldsymbol{\mu}, \boldsymbol{\nu})$ with*

- *$X = \{x_1, ..., x_n\}$ a set of real-valued* state variables *and $U, Y \subseteq X$ the set of* controllable *and* constrained *variables, respectively. We suppose that $Y = \{x_1, ..., x_m\}$;*
- *$\theta = \theta_1 \times ... \times \theta_n$, with $\theta_i = \{\theta_i^1, ..., \theta_i^{p_i}\}$ such that $0 < \theta_i^1 < ... < \theta_i^{p_i}$, associates with every dimension an ordered set of* thresholds*;*
- *$\boldsymbol{\mu} : \mathcal{D}^r(\theta) \to \mathbb{R}^m_{\geqslant 0}$ associates with every regulatory domain a vector of* production rates*;*
- *$\boldsymbol{\nu} : \mathcal{D}^r(\theta) \to diag(\mathbb{R}^m_{>0})$ associates with every regulatory domain a diagonal matrix of* degradation rates*.*

The set X is partitioned into a set U of *controllable* and a set $X \setminus U$ of *uncontrollable* variables. In addition, we suppose X to be partitioned into a set Y of *constrained* and a set $X \setminus Y$ of *unconstrained* or *input* variables. Both partitions are independent. For $D \in \mathcal{D}$, let $R(D) \subseteq \mathcal{D}^r$ be the set of regulatory domains that have D in their boundary, such that $R(D) = \{D\}$ for $D \in \mathcal{D}^r$.

In regulatory domains, that is for states $\mathbf{x} \in D$ for some $D \in \mathcal{D}^r$, the behavior of M is given by (2). In switching domains, where (2) is not defined, Filippov [16] defines the possible behaviors by the differential inclusion $\dot{\mathbf{x}} \in H(\mathbf{x})$ with

$$H(\mathbf{x}) = \bar{co}\big(\{\boldsymbol{\mu}(D') - \boldsymbol{\nu}(D')\mathbf{x} \mid D' \in R(D)\}\big)$$

where $\bar{co}(E)$ is the smallest closed convex hyper-rectangle containing the set E [16, 13]. Thus, for $\mathbf{x} \in D$ for some regulatory domain $D \in \mathcal{D}^r$, $H(\mathbf{x}) = \{\boldsymbol{\mu}(D) - \boldsymbol{\nu}(D)\mathbf{x}\}$.

Definition 16 (Semantics) *The trajectories of M are the solutions of $\dot{\mathbf{x}} \in H(\mathbf{x})$.*

Within a regulatory domain D with $\dot{\mathbf{x}} = \boldsymbol{\mu}^D - \boldsymbol{\nu}^D\mathbf{x}$, the protein concentrations evolve towards a *target equilibrium*, solution of $\mathbf{0} = \boldsymbol{\mu}^D - \boldsymbol{\nu}^D\mathbf{x}$:

Definition 17 ($\boldsymbol{\phi}$) *For any $D \in \mathcal{D}^r$, let $\boldsymbol{\phi}(D)$ denote the* target equilibrium *of D such that $\phi_i = \mu_i(D)/\nu_i(D)$ for $i = 1, ..., m$ (the constrained variables).*

Hypothesis 1 *Throughout this paper we make the assumption that for any regulatory domain D, $\exists D' \in \mathcal{D}^r$. $\boldsymbol{\phi}(D) \in D'$, that is, all target equilibria lie within regulatory domains, as in [13].*

2.3.2 Qualitative model

In this section we shortly present the qualitative model of a given piecewise linear system, as defined in [13].

Definition 18 (eq) *Given $\theta = \theta_1 \times ... \times \theta_n$, we define predicates $eq_i^{\#}$ on $\mathcal{D}$, $i \in \{1, ..., n\}$, $\# \in \{<, \leqslant, \geqslant, >\}$ such that*

- *if $x_i \in Y$ then*

$$eq_i^{\#}(D) \iff \bigvee_{D' \in R(D)} \forall x_i \in D'_i \,.\, \phi_i(D') \# x_i$$

  for $\# \in \{<, >\}$ and any $D = (D_1, ..., D_n) \in \mathcal{D}$, and $eq_i^{\leqslant} = \neg eq_i^{>}$, and $eq_i^{\geqslant} = \neg eq_i^{<}$;

- *if $x_i \in X \setminus Y$ then $eq_i^{\#} = true$.*

It can be seen that the predicates on switching domains are recursively defined with respect to lower-order switching domains. Intuitively, the predicates $eq_i^{\#}$ "interpolate" the relative position of target equilibria of the adjacent regulatory domains. The predicates $eq_i^{<}(D)$ and $eq_i^{>}(D)$ specify when some adjacent regulatory domain has its equilibrium left of D_i and right of D_i, respectively.

Let reg_i and $switch_i$ be the predicates characterizing the regulatory and switching domains of $\mathcal{D}_i$, respectively. For $D \in \mathcal{D}$, let $switching(D) = \{x_i \mid switch_i(D)\}$ be the set of switching variables in D. Let $succ_i$ and $prec_i$ be the successor and predecessor function on the ordered set of domains $\mathcal{D}_i$.

Definition 19 *Given a piecewise linear system $M = (X, U, Y, \theta, \boldsymbol{\mu}, \boldsymbol{\nu})$ of dimension n, the qualitative model $Q(M)$ is defined as the state-transition graph $Q(M) = (\mathcal{D}, \rightarrow)$ with transitions $\rightarrow \subseteq \mathcal{D} \times \mathcal{D}$ such that $\forall D, D' \in \mathcal{D}$,*

$$\begin{aligned} D \rightarrow D' \iff & switching(D) \subset switching(D') \wedge not\_tr(D) \wedge \\ & \bigwedge_{i=1,...,n} \Big(D'_i = prec_i(D_i) \wedge eq_i^{<}(D) \;\vee\; D'_i = succ_i(D_i) \wedge eq_i^{>}(D) \;\vee\; D'_i = D_i \Big) \vee \\ & switching(D') \subset switching(D) \wedge not\_tr(D') \wedge \\ & \bigwedge_{i=1,...,n} \Big(D'_i = prec_i(D_i) \wedge eq_i^{\leqslant}(D') \;\vee\; D'_i = succ_i(D_i) \wedge eq_i^{\geqslant}(D') \;\vee\; D'_i = D_i \Big) \end{aligned}$$

where

$$not\_tr(D) = \bigwedge_{x_i \in Y} \big(reg_i(D) \;\vee\; \min_{D' \in R(D)} \phi_i(D') < D_i < \max_{D' \in R(D)} \phi_i(D') \big)$$

Intuitively, condition $not\_tr(D)$ is satisfied if domain D is not *transient*, in the sense that it is instantaneously crossed by any trajectory of M reaching D. For the case where $Y = X$, $Q(M)$ corresponds exactly to the qualitative transition graph defined in [13].

2.3.3 Component Models

We now propose the construction of a pair of component-based models from a piecewise linear system. They are symbolic, in the sense that the state space and subsets thereof are represented as predicates, rather than by enumerating the states.

Proposition 1 $not\_tr = \bigwedge_{i=1,...,n} \left(reg_i \ \vee \ inc_i(eq_i^{\leqslant}) \wedge dec_i(eq_i^{\geqslant}) \ \vee \ eq_i^{<} \wedge eq_i^{>}\right)$.

Let $\mathcal{P}(X)$ denote the set of predicates on X.

Definition 20 ($C(M)$) *Given a piecewise linear system $M = (X, U, Y, \theta, \boldsymbol{\mu}, \boldsymbol{\nu})$, we define the component model $C(M) = (Levels, A, A^c, G, F) = (B_1 \| B_2 \| ... \| B_n)/V$ as follows.*

- $\forall i = 1, ..., n \ . \ B_i = \texttt{counter}(\mathcal{D}_i)$, *and*
- $V : A \to \mathcal{P}(\mathcal{D}(\theta))$ *such that* $V(B_i.inc) = V_i^{>}$ *and* $V(B_i.dec) = V_i^{<}$ *with*

$$V_i^{<} = reg_i \wedge not\_tr \wedge eq_i^{<} \vee \tag{3}$$

$$switch_i \wedge inc_i(not\_tr \wedge eq_i^{\leqslant}) \tag{4}$$

$$V_i^{>} = reg_i \wedge not\_tr \wedge eq_i^{>} \vee \tag{5}$$

$$switch_i \wedge dec_i(not\_tr \wedge eq_i^{\geqslant}) \tag{6}$$

where $\texttt{counter}(\mathcal{D}_i)$ *is a bounded counter defined on* $\mathcal{D}_i(\theta)$ *by the transition system* $\texttt{counter}(\mathcal{D}_i) = \big(\{level_i\}, \{inc_i, dec_i\}, \{G^{inc_i} = level_i \leqslant \theta_i^{p_i}, G^{dec_i} = level_i \geqslant \theta_i^1\}, \{F^{inc_i} = succ_i(level_i), F^{dec_i} = prec_i(level_i)\}\big)$ *where* $\{inc_i, dec_i\}$ *are controllable (resp. constrained) actions if* $x_i \in U$ *(resp.* $x_i \in Y$*).*
Let $(Q, \rightsquigarrow)$ *be the semantics of* $C(M)$.

Actions inc_i (dec_i) correspond to an increase (decrease) by one of the discretized concentration $level_i$ of protein i. Notice that we do not assume unconstrained variables to be constant. The predicates $V_i^{<}$ and $V_i^{>}$ specify when a transition decrementing x_i and incrementing x_i, respectively, is enabled. More precisely, lines (3) and (5) specify that there is a transition from a lower-order to a higher-order switching domain in the direction of the target equilibrium of the source domain. Lines (4) and (6) give the conditions for transitions decreasing the order: they must be compatible with the relative position of the target equilibrium of the destination domain. Condition not_tr makes sure that there is a transition from a switching domain $D \in \mathcal{D}^s$ to a higher-order switching domain only if D is not transient for any dimension; there is a transition to a lower-order domain D only if D is not transient for any dimension. Notice that $V_i^{<} \wedge V_i^{>}$ is not necessarily *false*, since switching domains may have more than one target equilibrium.

Remark 1 *Since $\|$ is associative, the above definition leaves open how the system is actually partitioned into components. The two extreme cases are that each B_i is considered as one component, or that $B_1 \| B_2 \| ... \| B_n$ is considered as one single component. This choice will usually depend on the degree of interaction between the modeled proteins. Putting all proteins in one component amounts to a non-modular model leading to non-compositional analysis. Representing each protein with a separate component may lead to a too heavy abstraction of the behavior. A good choice may gather closely interacting proteins, for instance proteins in the same cell, in one component, while modeling neighboring cells as separate components.*

We define a second component model over-approximating the transition relation of $Q(M)$.

Definition 21 ($\hat{C}(M)$) *Given a piecewise linear system M, we define the component model $\hat{C}(M) = (\mathcal{D}, A, A^c, G, F) = (B_1 \| B_2 \| ... \| B_n)/V$ as follows.*

- $\forall i = 1, ..., n \ . \ B_i = \texttt{counter}(\mathcal{D}_i)$, *and*
- $V : A \to \mathcal{P}(\mathcal{D})$ *such that* $V(B_i.inc) = V_i^{>}$ *and* $V(B_i.dec) = V_i^{<}$ *with*

$$\hat{V}_i^{<} = reg_i \wedge not\_tr \wedge eq_i^{<} \ \vee \ switch_i \wedge inc_i(eq_i^{\leqslant})$$

$$\hat{V}_i^{>} = reg_i \wedge not\_tr \wedge eq_i^{>} \ \vee \ switch_i \wedge dec_i(eq_i^{\geqslant})$$

Let $(Q, \hat{\rightsquigarrow})$ *be the semantics of* $\hat{C}(M)$.

Example 1 *Let us consider an example of two proteins* a *and* b *inhibiting each other's production [13]. Their respective production rates are defined by*

$$\mu_a = \begin{cases} 20 & \text{if } 0 \leqslant x_a < \theta_a^2 \wedge 0 \leqslant x_b < \theta_b^1 \\ 0 & \text{otherwise} \end{cases}$$

$$\mu_b = \begin{cases} 20 & \text{if } 0 \leqslant x_a < \theta_a^1 \wedge 0 \leqslant x_b < \theta_b^2 \\ 0 & \text{otherwise} \end{cases}$$

with $\theta_a^1 = \theta_b^1 = 4$ *and* $\theta_a^2 = \theta_b^2 = 8$. *The degradation rates of both proteins are always* 2. *Figure 1 sketches the dynamics of the system, and shows the corresponding qualitative model* $Q(M)$ *as defined in [13]. Figure 2 shows the semantics of* $C(M)$ *and* $\hat{C}(M)$.

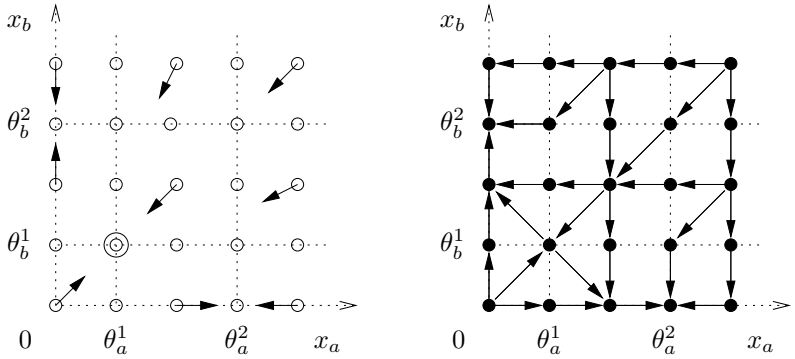

Figure 1: Phase space of the piecewise linear system with relative position of target equilibria (left) and transition system of the qualitative model $Q(M)$ (right).

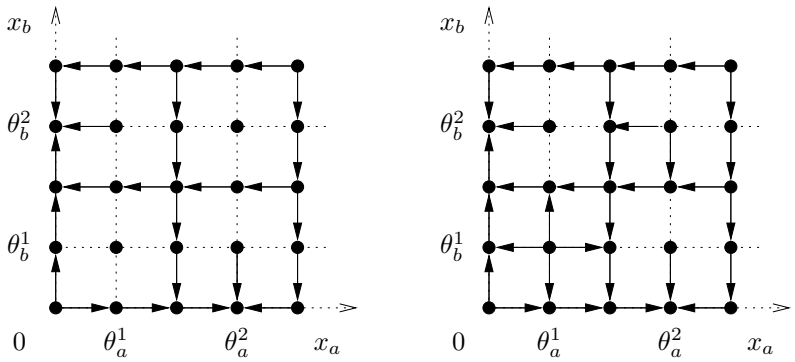

Figure 2: Component models $C(M)$ (left) and $\hat{C}(M)$ (right).

2.3.4 Model Comparison

The following two theorems show that $C(M)$ faithfully models $Q(M)$, up to the missing diagonal transitions. From a biological point of view, this approximation can be justified by the fact that production and degradation of proteins are never precisely measured and represented by the differential equations, nor are they truly continuous, but (on a molecular level) discrete functions. Therefore, any correct (that is, biologically sound) model will be robust enough so as to allow for interleaving transitions instead of synchronous steps.

Theorem 1 *If* $D \rightsquigarrow D'$ *for* $D, D' \in \mathcal{D}$, *then* $D \to D'$.

That is, if $C(M)$ can make some transition, then $Q(M)$ can make the same transition.

Proof. Suppose that $D \rightsquigarrow D'$. Further suppose, without loss of generality, that $D'_k < D_k$, and $\forall i \neq k \,.\, D'_i = D_i$. We distinguish two cases.

If $switch(D'_k)$ then by Definition 20, $V_k^<(D)$ and thus $(not\_tr \wedge eq_k^<)(D)$. By definition of $\rightarrow$ it follows that $D \rightarrow D'$.

If on the other hand, $switch(D_k)$, then $\left(inc_k(eq_k^\leqslant)\right)(D)$, and thus $eq_k^\leqslant(D')$. Define D'' such that for any $i = 1, ..., n$, $D''_i = prec_i(D_i)$ if $\left(switch_i \wedge inc_i(eq_i^\leqslant)\right)(D)$, $D''_i = succ_i(D_i)$ if $\left(switch_i \wedge dec_i(eq_i^\leqslant)\right)(D)$, and $D''_i = D_i$ otherwise. We have $D'_k = D''_k$ and $\forall i \,.\, reg_i(D'') \vee \left(inc_i(eq_i^>)\right)(D'') \wedge \left(dec_i(eq_i^<)\right)(D'')$. Thus by definition of $eq_i^<$ and $eq_i^>$, $\forall i \,.\, reg_i(D'') \vee (eq_i^> \wedge eq_i^<)(D'')$, and $not\_tr(D'')$. By definition of $\rightarrow$, it follows that $D \rightarrow D''$. Since $not\_tr(D')$, it follows that $D \rightarrow D'$. ■

Theorem 2 $\forall D, D' \in \mathcal{D} \,.\, D \rightarrow D' \implies D \rightsquigarrow D'$ *if* $\exists k \,.\, D_k \neq D'_k \wedge \forall j \neq k \,.\, D'_j = D_j$.

Intuitively, if $Q(M)$ can make some transition changing only the value of one variable, then $C(M)$ can make the same transition.

Proof. Suppose that $D \rightarrow D'$. Further suppose without loss of generality that $\forall i \,.\, D'_i \leqslant D_i$. We distinguish two cases: $switching(D) \subset switching(D')$ (increasing order) and $switching(D') \subset switching(D)$ (decreasing order). In the first case, $(not\_tr \wedge eq_k^<)(D)$ holds by Definition 19, and thus $V_k^<(D)$. In the second case, $eq_k^\leqslant(D')$ and thus $\left(inc_k(eq_k^\leqslant)\right)(D)$, and $V_k^<(D)$. In both cases, the implication follows. ■

When some diagonal (order-increasing or -decreasing) transition is possible in $Q(M)$, then some interleaving is possible in $\hat{C}(M)$. Formally:

Proposition 2 *If* $D \rightarrow D'$ *and* $switching(D') \subset switching(D)$, *then* $D \hat{\rightsquigarrow}^* D'$.

Proof. Suppose without loss of generality that $\forall i \,.\, D'_i \leqslant D_i$, and let $I = \{i \mid D_i \neq D'_i\}$. It follows by Definition 19 that $\bigwedge_{i \in I} eq_i^\leqslant(D')$. Let $(i_1, i_2, \ldots, i_k)$ be some permutation of the indices in I. Consider the sequence of domains $D^{i_0}, D^{i_1}, D^{i_2}, \ldots, D^{i_k} = D'$ such that $D^{i_0} = D$, and for any $j \in \{1, ..., k\}$, D^{i_j} differs from $D^{i_{j-1}}$ only in the fact that variable x_{i_j} is switching in $D^{i_{j-1}}$ and regulatory in D^{i_j}. By definition of $\hat{V}^<$ it follows that $D \hat{\rightsquigarrow} D_{i_1} \hat{\rightsquigarrow} \ldots \hat{\rightsquigarrow} D_{i_{k-1}} \hat{\rightsquigarrow} D'$. ■

This proof even shows the stronger result that *any* interleaving of an order-decreasing transition of $Q(M)$ is possible in $C(M)$.

2.3.5 Under-approximation

As a simplification of the component model, we propose a restriction to regulatory and first-order switching domains. As we will show, the under-approximation is often still precise enough to check reachability properties.

Definition 22 ($\mathcal{D}^{01}$) *Let* $\mathcal{D}^{01}$ *characterize the set of regular domains and first-order switching domains:*

$$\mathcal{D}^{01} = \mathcal{D} \wedge \bigvee_{i \in \{1,\ldots,n\}} \bigwedge_{j \in \{1,\ldots,n\} \setminus \{i\}} reg_j$$

Under-approximation is achieved by restricting the model with the state constraint $\mathcal{D}^{01}$. That is, we consider $C(M)/\mathcal{D}^{01}$ instead of $C(M)$. This is clearly an under-approximation of the behavior defined by the qualitative model.

3 Compositional Symbolic Analysis

In this section we discuss results to compositionally check reachability properties and existence of equilibrium states. For each of these properties, a sufficient condition is defined that implies the property. In order to ensure efficiency without being too conservative, the sufficient conditions use both the under-approximation $C(M)$ and the over-approximation $\hat{C}(M)$ of the piecewise linear system M.

3.1 Reachability

The following definitions and result about reachability are taken from [18].

In the sequel we consider a system $B = (A, X, G, F) = (B_1 \| \ldots \| B_N)/V$ with $B_i = (A_i, X_i, G_i, F_i)$, $i \in K = \{1, \ldots, N\}$, and V a restriction. That is, we suppose the n proteins to be modeled with N ($1 \leqslant N \leqslant n$) components, according to Remark 1. For $c \in \mathcal{P}(X)$ a conjunction of literals on X and $k \in K$, let $c|_k$ be the projection (abstraction) of c on X_k (thus, $c = \bigwedge_{k \in K} c|_k$). For $P \in \mathcal{P}(X)$ a predicate on X, let $P\bullet = \{a \in A \mid \exists \mathbf{x} \,.\, \big(P \wedge G^a(\mathbf{x})\big) \wedge \neg P\big(F^a(\mathbf{x})\big)\}$ be the set of actions leaving P, and let $c(P)$ be the set of prime implicants of P (thus, $\bigvee_{c \in c(P)} c = P$).

Let $s_k : \mathcal{P}(X_k) \times \mathcal{P}(X_k) \to 2^{A_k}$ be a local scheduler of k telling which action to take in order to get from some current state towards a destination state. This scheduler can be computed locally: for any predicate $P \in \mathcal{P}(X)$ and state $\mathbf{x}$ such that $\big(PRE_k(P)\big)(\mathbf{x})$, let

$$s_k(\mathbf{x}, P) = \begin{cases} \emptyset & \text{if } P(\mathbf{x}) \\ \{a \in A_k^c \mid \big(pre_a(pre\_c^{j-1})(P)\big)(\mathbf{x})\} & \text{otherwise} \end{cases}$$

where $j = \min\{i \mid \big(pre\_c^i(P)\big)(\mathbf{x})\}$ is the number of steps to steer B from $\mathbf{x}$ to P. Let $d_k(\mathbf{x}, P) = j$ denote this distance, and let $d(\mathbf{x}, P) = \Sigma_{k \in K} d_k(\mathbf{x}, P)$. Intuitively, $s_k(\mathbf{x}, P)$ is the set of actions that, when executed in state $\mathbf{x}$, bring B_k closer to P, while keeping the component within the controllable predecessors $PRE_k(P)$. In computing s_k we assume that components are sufficiently small to allow local controller synthesis; otherwise large components have to be further decomposed, or an abstraction be used to compute the schedulers.

Let $s : \mathcal{P}(X) \times \mathcal{P}(X) \to 2^A$ be the "union" of the local schedulers such that $s(\mathbf{x}, \mathbf{y}) = \bigcup_{k \in K} \{s_k(\mathbf{x}|_k, \mathbf{y}|_k)\}$. Let $s_k(P) = \bigcup_{\mathbf{x}} s_k(\mathbf{x}, P)$, and $s(P) = \bigcup_{\mathbf{x}} s(\mathbf{x}, P)$ be the set of all actions used by s to reach P.

Definition 23 (Unblocking order) *For $a \in A$, $c \in \mathcal{P}(X)$ a conjunction on literals in X, and $k \in K$, let*

$$pr_{sched}^{a,c,k} = \bigoplus_{i \geqslant 1} \big(C \mapsto \{a \prec b \mid a, b \in A_k^c \ \wedge \ pre_a(\neg C') \wedge pre_b(C') \neq \mathit{false}\}\big)$$

where $C = blocked(a) \wedge pre\_c_k^i(c|_k)$ and $C' = pre\_c_k^{i-1}(c|_k)$.

The unblocking order $pr_{sched}^{a,c,k}$ resolves conflicts within component B_k, by giving priority to actions leading towards c over actions leading away from c, in order to unblock a.

Definition 24 (Unblocking schedule) *Given $a \in A^c$ with $(G^a)' \neq G^a$, we call*

$$unblock(a) = \Big\{ \Big(c,\ \big(blocked(a) \mapsto c\bullet \cup \{a\} \prec s(c)\big) \oplus \bigoplus_{k \in K} pr_{sched}^{a,c,k} \Big) \mid c \in c\big((G^a)'\big) \wedge c\bullet \subseteq A^c \Big\}$$

the set of unblocking schedules *of a.*

An unblocking schedule for some action a is thus a tuple of a (convex) predicate in which a is enabled, and a dynamic priority. The latter is composed of an unblocking order for each component, and a dynamic priority giving priority to scheduler actions leading towards c over actions leaving c.

An *unblocking domain* specifies the states for which unblocking schedules are defined:

Definition 25 (Unblocking domain) *Given a conjunction c on X, let $reach(c) = \bigwedge_{k \in K} PRE_k(c|_k)$ be the* unblocking domain *of c.*

Algorithm 1 Algorithm enabling action a.

```
while ¬(G^a)'(x) do
   a' := choice (s*(x, a)^A);
   x := F^{a'}(x)
od
```

Consider some action $a \in A^c$. To effectively reach a state where a is enabled, we compute, for a current state $\mathbf{x}$, a sequence of scheduling decisions that converges towards the choice of an action that brings the system closer to a state enabling a. That is, if a is not enabled, fire some action a' moving the system towards a state

where a is enabled; if no such a' is enabled, fire some a'' bringing the system closer to a', and so on. In doing this, we have to disable actions increasing the distance of any component from c. This is what algorithm 1 does. $s^*(\mathbf{x},a)^A$ is the fixed-point of a sequence $[s^i(\mathbf{x},a)^A]_{i\geqslant 0}$, where any element $s^i(\mathbf{x},a)$ is a *scheduling decision*. It is a tuple $(c^i,\prec^i,A^i)$ consisting of a predicate to be reached, a priority to coordinate system execution so as to avoid conflicts with previous scheduling decisions $s^j(\mathbf{x},a)$ with $j<i$, and a set of actions to be scheduled in order to approach c^i. The sequence of scheduling decisions is defined as follows. $s^0(\mathbf{x},a)=\big(c,\ pr(\mathbf{x}),\ s(\mathbf{x},c)\big)$ for some $(c,pr)\in unblock(a)$ such that $reach(c)(\mathbf{x})$ and $\forall(c',pr')\in unblock(a)$ with $reach(c')(\mathbf{x})$, $d(\mathbf{x},c)\leqslant d(\mathbf{x},c')$ (that is, choose some optimal unblocking schedule). For any $i\geqslant 0$, let

$$s^{i+1}(\mathbf{x},a)=\begin{cases} s^i(\mathbf{x},a) & \text{if } legal\_enabled(a,i,\mathbf{x})\neq\emptyset \\ \big(s^0(\mathbf{x},a')^c,\prec,s^0(\mathbf{x},a')^A\big) & \text{otherwise}\end{cases}$$

where $legal\_enabled(a,i,\mathbf{x})=EA(\mathbf{x})/s^i(\mathbf{x},a)^{\prec}\cap s^i(\mathbf{x},a)^A$, is the set of actions to be scheduled that are enabled, $a'=choice(A/s^i(\mathbf{x},a)^{\prec}\cap s^i(\mathbf{x},a)^A)$, $\prec=s^i(\mathbf{x},a)^{\prec}\oplus s^0(\mathbf{x},a')^{\prec}$, and *choice* is an arbitrary (but deterministic) choice.

Remark 2 *The predicate* $\bigvee_{(c,pr)\in unblock(a)} reach(c)$ *under-approximates the* co-reachable *set of states from which any of the actions can be reached. A sufficient condition guaranteeing reachability of some state can be found in [18].*

In addition to a path from an initial to a final state, the nesting of the used unblocking schedules can be used to provide diagnostics about the causality between the executed actions. An example is given in the case study below.

Remark 3 *Let $P\in\mathcal{P}(X)$ be a predicate on $\mathbf{X}$. P is reachable if an observer component $o(P)=(\emptyset,\{alive\},\{G^{alive}=true\},\{F^{alive}=id_\emptyset\})$ with restriction $U^{alive}=P$ is deadlock-free in $\big(B\|o(P)\big)/U^{alive}$. Moreover, a scheduler constructed as above controls the system so that P is effectively reached.*

Notice that composing with observer $o(P)$ and restricting with U^{alive} do not change the behavior of the other components. Of course, if some action a already has guard $(G^a)'=P$, no observer needs to be added to check for reachability of P and construct a scheduler.

Clearly, reachability in the under-approximation implies reachability in the exact model:

Proposition 3 *For any* $\mathbf{x}$, $\mathbf{x}'$, $\mathbf{x}\rightsquigarrow_{\mathcal{D}^{01}}\mathbf{x}' \implies \mathbf{x}\rightsquigarrow\mathbf{x}'$.

Therefore, reachability can be checked efficiently on the under-approximation. If no path is found, the search can be refined by checking with the exact model.

3.2 Equilibrium States

Equilibrium states of a piecewise linear system are states in which "the system may remain". In regulatory domains, equilibrium states of a piecewise linear system $M=(X,U,Y,\theta,\boldsymbol{\mu},\boldsymbol{\nu})$ of dimension n are fixed points of (2). In the generalization of Filippov [16], the set of equilibrium states is defined as

$$\{\mathbf{x}\in\mathbb{R}^n_{\geqslant 0}\mid \mathbf{0}\in H(\mathbf{x})\}$$

We provide an efficient way to compute the set of equilibria, and approximations for the sets of stable equilibria and unstable equilibria as defined in [7].

Definition 26 (Qualitative equilibria) *We define the qualitative equilibrium states $\mathcal{EQ}$, stable equilibria $\mathcal{EQ}^s$, and unstable equilibrium states $\mathcal{EQ}^u$ by*

$$\mathcal{EQ}=\bigwedge_{x_i\in Y}\big(reg_i\wedge eq_i^{=}\ \vee\ switch_i\wedge inc_i(eq_i^{\leqslant})\wedge dec_i(eq_i^{\geqslant})\ \vee\ eq_i^{<}\wedge eq_i^{>}\big)$$

$$\mathcal{EQ}^s=\mathcal{D}\wedge\neg\bigvee_{x_i\in Y}\big(\hat{V}_i^{<}\vee\hat{V}_i^{>}\big)$$

$$\mathcal{EQ}^u=\mathcal{EQ}\wedge\neg\bigwedge_{x_i\in Y}\big(reg_i\wedge eq_i^{=}\ \vee\ switch_i\wedge inc_i(eq_i^{>})\wedge dec_i(eq_i^{<})\big)$$

Notice that $\mathcal{EQ}^s$ is computed on the over-approximation, whereas the other equilibria are computed from the under-approximation. The predicate $\mathcal{EQ}$ characterizes exactly the set of domains containing an equilibrium state, whereas $\mathcal{EQ}^s$ and $\mathcal{EQ}^u$ under-approximate the sets of stable and unstable equilibria, respectively:

Definition 27 (Ψ) *For any switching domain $D \in \mathcal{D}$ of order k, let*

$$\Psi(D) = C \cap \overline{rect}\big(\{\phi(D') \mid D' \in R(D)\}\big)$$

where C is the $(n-k)$-dimensional threshold hyperplane containing D, and $\overline{rect}(E)$ denotes the smallest closed hyper-rectangle containing the set E [13].

For a domain $D \in \mathcal{D}$, $\Psi(D)$ is a set over-approximating the set of of target equilibria of D.

Theorem 3

$$\forall D \in \mathcal{D} \,.\, \big(\mathcal{EQ}(D) \iff \Psi(D) \cap D \neq \emptyset\big)$$

Proof. $\mathcal{EQ}(D) \iff \bigwedge_{x_i \in Y} \big(reg_i \wedge eq_i^{=} \;\vee\; switch_i \wedge inc_i(eq_i^{\leqslant}) \wedge dec_i(eq_i^{\geqslant}) \;\vee\; eq_i^{<} \wedge eq_i^{>}\big) \iff \exists \mathbf{x} \in D \,.\, \bigwedge_{x_i \in Y} \min_{D' \in R(D)} \phi_i(D') \leqslant \mathbf{x} \leqslant \max_{D' \in R(D)} \phi_i(D') \iff \Psi(D) \cap D \neq \emptyset$. ■

Conjecture 1

$$\forall D \in \mathcal{D} \quad . \quad \mathcal{EQ}^{\#}(D) \implies D \cap EQ^{\#} \neq \emptyset$$

for $\# \in \{s, u\}$, where EQ^s and EQ^u denote the stable and unstable equilibrium sets of M [7].

Example 2 *Figure 3 shows the equilibria found for Example 1. There are three qualitative equilibrium states, two of which are stable. There is no unstable equilibrium. This is the exact result, that is, identical to the equilibria obtained by using the criteria of [7].*

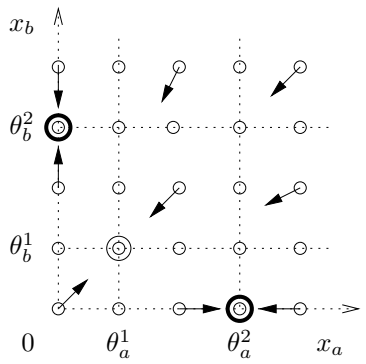

Figure 3: Equilibria (double circles) and stable equilibria (fat double circles).

3.3 Nested Properties

According to remark 3, it is possible to check reachability of an arbitrary predicate, and thus to check for properties like reachability of equilibrium states. Examples are considered in the following case studies.

4 Case Studies

4.1 The Sporulation of Bacillus subtilis

We have experimented the above modeling and verification techniques with a well-known example, the initiation of sporulation of B. subtilis. The model of the genetic network in the form of piecewise linear differential equations is taken from [11]. It consists of 11 interacting proteins, 2 of which are unconstrained inputs. The differential equations distinguish up to 9 qualitative concentration levels for each protein.

System 1 PROMETHEUS model of the initiation of sporulation of B. subtilis (extract).

```
SYSTEM b_subtilis

    // component declarations:
    COMPONENT AbrB
        INT x(4);
        TRANSITIONS
            ACTION inc_x IF x<=3 DO x++;
            ACTION dec_x IF x>=1 DO x--;
    END AbrB
    [...]

    // declarations of target equilibria:
    LET k_AbrB_2 = SigA.H AND AbrB.x<3 AND !(SpoA.x>1 AND Signal.H);
    EQUILIBRIUM AbrB.x = 0 IF !$k_AbrB_2;
    EQUILIBRIUM AbrB.x = 4 IF $k_AbrB_2;
    [...]

END b_subtilis
```

The algorithms for reachability and stable states have been integrated in the compositional verification tool PROMETHEUS. For the sake of convenience, we have added primitives to the input language of PROMETHEUS to specify the target equilibria of the regulatory domains similar to the input language of GNA [12]. An extract of the model is shown in System 1 (for each protein i, the intervals in $\mathcal{D}_i$ representing the discretized level of x_i are numbered from 0 to $|\mathcal{D}_i| - 1$). All actions are supposed to be controllable for this example. For this case study we suppose all input concentrations to be fixed, so as to be able to compare with the results of GNA. Each protein is modeled as one component.

In both $C(M)$ and the under-approximation $C(M)/\mathcal{D}^{01}$ restricted to regulatory and first-order switching domains, the same path of length 15 from the initial to the final state is found: `< < AbrB.dec AbrB.dec > SigH.inc SigH.inc KinA.inc KinA.inc KinA.inc KinA.inc < Hpr.dec Hpr.dec > SinI.inc SinI.inc Spo0A.inc Spo0A.inc > SigF.inc`. That is, the level of `AbrB` decreases, which enables production of `SigH`; the decreasing concentration of `Hpr` enables the production of `SinI`. Finally, `SigF` is produced. This path is slightly longer than the shortest paths determined by GNA, which consist of 13 steps. This is because GNA twice increases both `KinA` and `Spo0A` in the same transition, whereas in $C(M)$, both actions have to be serialized. Therefore, the path found by PROMETHEUS is a shortest path in $C(M)$.

Let us now examine the equilibria of the model. According to Definition 26, there exists one stable equilibrium, which corresponds to a state of non-sporulation of B. subtilis. No unstable equilibrium is found. A comparison with GNA shows that there indeed are no other equilibria. We can now use PROMETHEUS to find a path from the initial state to the stable equilibrium. This result is consistent with the experimental observation that for the same inputs, the bacteria can evolve in two different ways, either initiating sporulation or entering a stable non-sporulating state.

	$C(M)$	equilibria	unbl. schedules	path
B. subtilis (under-approx.)	< 10 ms	0.01 s	0.06 s	0.03 s
B. subtilis (exact)	0.08 s	0.01 s	8.1 s	0.34 s
E. coli (under-approx.)	< 10 ms	0.01 s	0.04 s	< 10 ms
E. coli (exact)	0.03 s	< 10 ms	2.7 s	0.04 s
Delta-Notch 19 (under-approx.)	< 10 ms	0.14 s	1.3 s	1.3 s
Delta-Notch 37 (under-approx.)	0.05 s	2.7 s	7.7 s	5.0 s

Figure 4: Benchmarks for the different algorithms.

Table 4.1 shows the execution times for the models of B. subtilis and another example, the nutritional stress response of E. coli, which is taken from [5]. The benchmarks suggest the use of under-approximation as long as

the model remains precise enough to verify the desired properties. This is the case for all of the case studies we did so far. Notice that the unblocking schedules have to be pre-computed only once. It is therefore possible to obtain fast responses to an arbitrary number of reachability problems of the model.

4.2 Delta-Notch Cell Differentiation

Another well-studied example of a genetic network is cell differentiation by delta-notch lateral inhibition [22, 17]. Cell differentiation is an important step in embryonic development, as it causes initially uniform cells to assume different functions.

For each cell we consider the concentrations of two trans-membrane proteins, *Delta* and *Notch*. Following the model provided in [22], high concentrations of *Delta* and *Notch* inhibit each other's production within the same cell, and high *Delta* levels activate further *Delta* production. Furthermore, a high *Delta* level activates *Notch* production in the neighboring cells. Figure 5 illustrates these interactions.

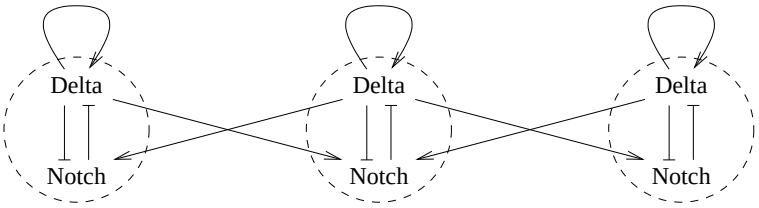

Figure 5: Interaction within and between neighbor cells.

For our case study, we consider a network consisting of 19 cells with the layout shown in Figure 6, and a network of 37 cells with a similar layout.

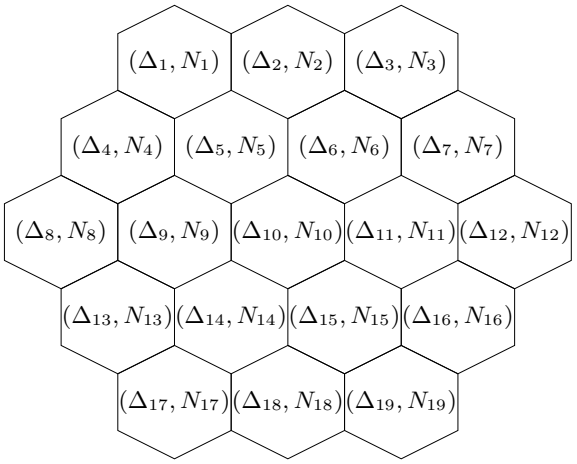

Figure 6: Model of 19 communicating cells.

For each protein we partition the continuous state space into two intervals and one threshold value: $\mathcal{D}_\Delta = \{[0, \theta_\Delta), \{\theta_\Delta\}, (\theta_\Delta, max_\Delta]\}$ and $\mathcal{D}_N = \{[0, \theta_N), \{\theta_N\}, (\theta_N, max_N]\}$. Cells with low *Delta* and high *Notch* levels ($\Delta < \theta_\Delta$, $Notch > \theta_N$) are undifferentiated, whereas cells with high *Delta* and low *Notch* concentrations ($\Delta > \theta_\Delta$,$Notch < \theta_N$) are differentiated. We are not interested in the actual production and degradation rates of the proteins but assume the target equilibria ϕ_{Δ_i} and ϕ_{Notch_i} to satisfy

$$\begin{aligned}
\phi_{\Delta_i} &< \theta_\Delta \text{ if } Notch_i > \theta_N \\
\theta_\Delta &< \phi_{\Delta_i} \leqslant max_\Delta \text{ if } Notch_i < \theta_N \\
\phi_{Notch_i} &< \theta_N \text{ if } \max\{\Delta_j \mid j \in neighbors(i)\} < \theta_\Delta \\
\theta_N &< \phi_{Notch_i} \leqslant max_N \text{ if } \max\{\Delta_j \mid j \in neighbors(i)\} > \theta_\Delta
\end{aligned}$$

Our model thus distinguishes three discrete states for each protein concentration. This abstraction reduces the represented interactions to a negative feedback from *Notch* to *Delta* within the same cell, and a positive feedback from *Delta* to the *Notch* production in neighboring cells, as in the formalization of [17]. As we will see, our model is still precise enough to reproduce the observed behavior, that is, the existence of equilibria and their reachability.

We choose to represent each cell by one component. Considering only regulatory and first-order switching domains for a system of n components, the $2n$-dimensional global state space encompasses 4^n regulatory domains and $2n \times 2^{2n-1}$ first-order switching domains, that is, 5.5×10^{12} states for 19 cells, and 7.2×10^{23} states for 37 cells.

PROMETHEUS finds that stable equilibria exist. They correspond to the 6861 possible configurations (in the case of 37 cells) where

- all cells are either differentiated or not differentiated,
- any differentiated cell has only non-differentiated direct neighbors, and
- any non-differentiated cell has at least one direct neighbor that is differentiated.

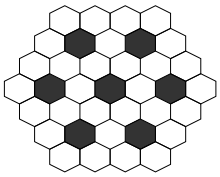

Figure 7: Example of a stable equilibrium state involving 37 cells (dark cells are differentiated).

Next, we check reachability of the stable states with 7 differentiated cells as shown in Figure 7, from the initial state where all cells are undifferentiated. Using algorithm 1, PROMETHEUS finds a path of length 28 within five seconds. The results reported by PROMETHEUS are consistent with the actual, experimentally observed behavior [22].

5 Discussion

We have presented a novel approach for component-based modeling and symbolic analysis of genetic regulatory networks. The model discretizes the network dynamics defined by a system of piecewise linear differential equations. It is then possible to compositionally verify reachability properties and automatically construct a path, and symbolically compute several kinds of equilibrium states. These results use sufficient conditions on both an under- and an over-approximation of the behavior of the genetic network, allowing to deal with complex, high-dimensional systems. Two case studies and several benchmarks show the potential of this approach.

Future work includes more complex case studies involving several communicating genetic networks of neighboring cells, to further evaluate the efficiency of our approach. We also intend to take advantage of the heterogeneous modeling capabilities of our component model [19] in order to study the modeling and analysis of interacting biological functions, such as genetic networks and metabolic pathways.

Acknowledgement. The author would like to thank Hidde de Jong and Grégory Batt (HELIX project, INRIA) for many fruitful discussions and comments on earlier versions of this work.

References

[1] K. Altisen, G. Gössler, and J. Sifakis. Scheduler modeling based on the controller synthesis paradigm. *Journal of Real-Time Systems, special issue on "control-theoretical approaches to real-time computing"*, 23(1/2):55–84, 2002.

[2] R. Alur, T. Dang, and F. Ivancić. Reachability analysis of hybrid systems via predicate abstraction. *Trans. on Embedded Computing Systems*, 2004.

[3] G. Batt, D. Bergamini, H. de Jong, H. Garavel, and R. Mateescu. Model checking genetic regulatory networks using GNA and CADP. In *proc. SPIN'04*, volume 2989 of *LNCS*, pages 158–163. Springer-Verlag, 2004.

[4] G. Batt, H. de Jong, J. Geiselmann, M. Page, D. Ropers, and D. Schneider. Symbolic reachability analysis of genetic regulatory networks using qualitative abstraction. Research Report 5362, INRIA, France, 2004.

[5] G. Batt, D. Ropers, H. de Jong, J. Geiselmann, M. Page, and D. Schneider. Qualitative analysis and verification of hybrid models of genetic regulatory networks: Nutritional stress response in escherichia coli. In M. Morari and L. Thiele, editors, *proc. HSCC'05*, volume 3414 of *LNCS*, pages 134–150. Springer-Verlag, 2005.

[6] G. Bernot, J.-P. Comet, A. Richard, and J. Guespin. Application of formal methods to biological regulatory networks: Extending Thomas' asynchronous logical approach with temporal logic. *Journal of Theoretical Biology*, 229(3):339–348, 2004.

[7] R. Casey, H. de Jong, and J.-L. Gouzé. Piecewise-linear models of genetic regulatory networks: Equilibria and their stability. Research Report 5353, INRIA, France, 2004.

[8] C. Chaouiya, E. Remy, P. Ruet, and D. Thieffry. Qualitative modelling of genetic networks: From logical regulatory graphs to standard petri nets. In J. Cortadella and W. Reisig, editors, *proc. ICATPN'04*, volume 3099 of *LNCS*, pages 137–156. Springer-Verlag, 2004.

[9] J.-P. Comet, H. Klaudel, and S. Liauzu. Modeling multi-valued genetic regulatory networks using high-level petri nets. In G. Ciardo and P. Darondeau, editors, *proc. ICATPN'05*, volume 3536 of *LNCS*, pages 208–227. Springer-Verlag, 2005.

[10] H. de Jong. Modeling and simulation of genetic regulatory systems: A literature review. *Journal of Computational Biology*, 9(1):69–105, 2002.

[11] H. de Jong, J. Geiselmann, G. Batt, C. Hernandez, and M. Page. Qualitative simulation of the initiation of sporulation in Bacillus subtilis. *Bulletin of Mathematical Biology*, 66(2):261–300, 2004.

[12] H. de Jong, J. Geiselmann, C. Hernandez, and M. Page. Genetic Network Analyzer: Qualitative simulation of genetic regulatory networks. *Bioinformatics*, 19(3):336–344, 2003.

[13] H. de Jong, J.-L. Gouzé, C. Hernandez, M. Page, T. Sari, and J. Geiselmann. Qualitative simulation of genetic regulatory networks using piecewise-linear models. *Bulletin of Mathematical Biology*, 66:301–340, 2004.

[14] S. Eker, M. Knapp, K. Laderoute, P. Lincoln, and C. Talcott. Pathway logic: Executable models of biological networks. *ENTCS*, 71, 2002.

[15] F. Fages, S. Soliman, and N. Chabrier-Rivier. Modelling and querying interaction networks in the biochemical abstract machine BIOCHAM. *J. of Biological Physics and Chemistry*, 4(2):64–72, 2004.

[16] A.F. Filippov. Differential equations with discontinuous righthand side. *Mathematics and its Applications*, 18, 1988.

[17] R. Ghosh, A. Tiwari, and C. Tomlin. Automated symbolic reachability analysis; with application to delta-notch signaling automata. In O. Maler and A. Pnueli, editors, *proc. HSCC'03*, volume 2623 of *LNCS*, pages 233–248. Springer-Verlag, 2003.

[18] G. Gössler. Reach scheduling for embedded systems. Research Report 5651, INRIA, France, 2005.

[19] G. Gössler and J. Sifakis. Composition for component-based modeling. *Science of Computer Programming*, 55(1-3):161–183, 2005.

[20] M. Heiner and I. Koch. Petri net based model validation in systems biology. In J. Cortadella and W. Reisig, editors, *proc. ICATPN'04*, volume 3099 of *LNCS*, pages 216–237. Springer-Verlag, 2004.

[21] B. Lewin. *Genes VIII*. Prentice Hall, 2003.

[22] G. Marnellos, G.A. Deblandre, E. Mjolsness, and G. Kintner. Delta-Notch lateral inhibitory patterning in the emergence of ciliated cells in *Xenopus*: Experimental observations and a gene network model. In R.B. Altman, K. Lauderdale, A.K. Dunker, L. Hunter, and T.E. Klein, editors, *Pacific Symposium on Biocomputing, PSB 2000*, volume 5, pages 326–337. World Scientific Publishing, 2000.

[23] O. Resendis-Antonio, J.A. Freyre-González, R. Menchaca-Méndez, R.M. Gutiérrez-Ríos, A. Martínez-Antonio, C. Avila-Sánchez, and J. Collado-Vides. Modular analysis of the transcriptional regulatory network of e. coli. *Trends in Genetics*, 21(1):16–20, 2005.

Unité de recherche INRIA Rhône-Alpes
655, avenue de l'Europe - 38334 Montbonnot Saint-Ismier (France)

Unité de recherche INRIA Futurs : Parc Club Orsay Université - ZAC des Vignes
4, rue Jacques Monod - 91893 ORSAY Cedex (France)
Unité de recherche INRIA Lorraine : LORIA, Technopôle de Nancy-Brabois - Campus scientifique
615, rue du Jardin Botanique - BP 101 - 54602 Villers-lès-Nancy Cedex (France)
Unité de recherche INRIA Rennes : IRISA, Campus universitaire de Beaulieu - 35042 Rennes Cedex (France)
Unité de recherche INRIA Rocquencourt : Domaine de Voluceau - Rocquencourt - BP 105 - 78153 Le Chesnay Cedex (France)
Unité de recherche INRIA Sophia Antipolis : 2004, route des Lucioles - BP 93 - 06902 Sophia Antipolis Cedex (France)

Éditeur
INRIA - Domaine de Voluceau - Rocquencourt, BP 105 - 78153 Le Chesnay Cedex (France)
http://www.inria.fr
ISSN 0249-6399